

TRABAJO DE FIN DE GRADO

Grado en Ingeniería Mecánica

**MONITORIZACIÓN DE LA INTEGRIDAD
ESTRUCTURAL DE UN AEROGENERADOR A ESCALA
MEDIANTE MÉTODOS BASADOS EN DATOS**



Memoria y anexos

Autor: Alejandro Martínez Berbel

Director: Francesc Pozo Montero

Departamento MAT

Co-Director: Yolanda Vidal Segui

Convocatoria: Junio 2018



RESUMEN

Este proyecto trata sobre la puesta a punto del prototipo de una torre eólica offshore para, posteriormente, realizar los tests pertinentes para la detección de algún tipo de daño en la estructura. En primer lugar se describe el marco teórico de la energía eólica, en concreto los parques offshore. Posteriormente se detalla paso a paso el montaje y puesta a punto de todo el instrumental necesario a la hora de recabar los datos, los cuales usaremos a la hora de crear el modelo de PCA – Análisis de componentes principales- .

En definitiva, se equipa la torre prototipo con todo el material necesario para recopilar -de forma experimental- el comportamiento dinámico de la estructura ante las vibraciones producidas por el movimiento de las aspas, las cuales son accionadas por la energía cinética del viento.

RESUM

Aquest projecte tracta sobre la posada a punt del prototip d'una torre eòlica offshore per, posteriorment, realitzar els tests pertinents per a la detecció d'algun tipus de dany en l'estructura. En primer lloc es descriu el marc teòric de l'energia eòlica, en concret els parcs offshore. Posteriorment es detalla pas a pas el muntatge i posada a punt de tot l'instrumental necessari a l'hora de recaptar les dades, les quals farem servir a l'hora de crear el model de PCA -Anàlisi de components principals-.

En definitiva, s'equipa la torre prototip amb tot el material necessari per recopilar -de forma experimental- el comportament dinàmic de l'estructura davant les vibracions produïdes pel moviment de les aspes, les quals són accionades per l'energia cinètica del vent.

ABSTRACT

This project is based on the development of the prototype of an offshore wind tower to subsequently carry out the tests indicated for the detection of some type of damage to the structure. First, the theoretical framework of wind energy is described, specifically offshore parks. Subsequently detailed step by step the assembly and tuning of all the necessary instruments at the time of collecting the data, which we will use when creating the PCA model-Principal Components Analysis-.

In short, the prototype tower is equipped with all the necessary material to collect -in an experimental way- the dynamic behavior of the structure when vibrations are produced by the movement of the blades, which are activated by the kinetic energy of the wind.



AGRADECIMIENTOS

En primer lugar quiero agradecer a mis padres y a mi hermano, quienes me han apoyado a lo largo de todo el grado, en especial durante este trabajo.

También a mis tutores, Yolanda Vidal y Francesc Pozo, por guiarme durante el proceso de realización del proyecto. Sin olvidar a David Agís, por su ayuda con el PCA.

ÍNDICE

RESUMEN	3
RESUM.....	4
ABSTRACT	5
AGRADECIMIENTOS.....	7
ÍNDICE.....	8
1. PREFACIO.....	11
1.1. Motivación	11
1.2. Alcance del proyecto	11
2. MARCO TEÓRICO.....	13
2.1. Energía Eólica	13
2.2. Impacto actual.....	14
2.3. Implantación al medio marino	15
2.4. Parques offshore	16
2.5. Detección de daños.....	18
3. HARDWARE	20
3.1. Torre prototipo.....	20
3.2. Acelerómetros.....	22
3.3. Devices NI	24
3.4. Excitador y amplificador de potencia.....	25
3.5. Generador de funciones.....	25
4. SOFTWARE.....	27
4.1. NI-Max.....	27
4.2. Matlab 2017.b	27
4.2.1. Analog Input Recorder	28
5. OBTENCION DE DATOS.....	30
5.1. Ensayos estructura sana (30 iteraciones).....	30
5.2. Ensayos sustituyendo una barra por otra igualmente sana (15 iteraciones)	31
5.3. Ensayos sustituyendo por una barra seccionada 5mm (15 iteraciones).....	31
5.4. Ensayos quitando un tornillo (15 iteraciones)	32
6. PROCESAMIENTO DE DATOS (PCA)	33
6.1. Volcado de datos.....	34

6.2. Código PCA	35
6.3. Nubes de puntos	37
6.4. Discusión de los resultados	41
CONCLUSIONES	42
BIBLIOGRAFÍA/WEBGRAFÍA.....	44
ANEXOS	46
ANEXO 1: Proceso de montaje del equipo	46
A1.1. Torre prototipo	46
A1.2. Acelerómetros y Devices NI	47
A1.3. Excitador, amplificador y generador de funciones	49
ANEXO 2: Código Matlab.....	50
A2.1. Sección 1	50
A2.2. Sección 2	50
A2.3. Sección 3	52
A2.4. Sección 4	53
A2.5. Sección 5	53
ANEXO 3: Gráficos obtenidos.....	54
A3.1. Estructura original.....	54
A3.2. Barra seccionada 5mm.....	55
A3.3. Barra sin un tornillo	57



1. PREFACIO

1.1. Motivación

En los últimos años la sociedad moderna ha aceptado que gran parte de sus fuentes de energía, utilizadas desde hace mucho tiempo, como los combustibles fósiles, no son un método sostenible de abastecimiento. Las reservas de estas fuentes son finitas, y el aumento de la población requiere de una mayor cantidad de energía para abastecerse.

Esta búsqueda de un abastecimiento sostenible de energía ha fomentado un auge en las energías renovables, las cuales tratan de crear energía a través del aprovechamiento no invasivo de fenómenos naturales. Energías tales como: solar, geotérmica, mareomotriz o **eólica** permiten extraer, con un impacto mínimo en la naturaleza, cantidades suficientes de energía para sustentar incluso grandes ciudades.

De igual forma, este auge ha llevado a los investigadores a buscar los mejores lugares donde extraer dichas energías. Estos lugares no siempre son los más idóneos para crear las estructuras e infraestructuras necesarias en el proceso de aprovechamiento de la energía.

El análisis de este trabajo se centra en estudiar la **integridad de la estructura de un molino eólico ante las vibraciones** producidas por el movimiento que causa el viento sobre las aspas. Dicho molino se encuentra **ubicado en el océano**, donde el agua salada y la fauna marina lo convierten en un medio bastante hostil para la integridad de su infraestructura.

1.2. Alcance del proyecto

En vista de satisfacer los requisitos necesarios para el correcto funcionamiento de una torre eólica ubicada en el océano, este proyecto realiza la preparación completa del prototipo, en vista de estudiar la respuesta dinámica de la estructura ante posibles **fallos o defectos** en zonas concretas de la parte sumergida, como barras seccionadas o pérdida de algún perno.

Estos fallos estructurales pueden llegar a considerarse significativos, ya que, tanto las vibraciones producidas por el movimiento de las aspas, como el encontrarse en un medio acuático y con alta salinidad, son factores a tener en cuenta a la hora de construir y mantener este tipo de estructuras, las cuales deben perdurar durante un largo tiempo.

Este trabajo parte de la base de un prototipo a escala de un molino eólico con una estructura inferior similar a la utilizada en los parques eólicos offshore. En lugar de las aspas, en la parte superior de la torre, un excitador simula las vibraciones que produciría el molino en funcionamiento.

Gracias a estas vibraciones podemos analizar la salud de la estructura bajo condiciones de trabajo. Los datos se recaban con la ayuda de 8 acelerómetros digitales, instalados en puntos concretos de la estructura. En vista de ceñirnos al máximo a un caso real, llevamos a cabo las pruebas con tres amplitudes distintas (rachas de viento) y sustituyendo partes sanas de la estructura por recambios con defectos (deterioro de la estructura), en vista de poder detectar estos fallos a través de los datos recibidos.

Los datos proporcionados por los acelerómetros se computan mediante Matlab, que los recoge en tablas de voltaje respecto al tiempo. Al procesar estos datos mediante el **Análisis de Componentes Principales** –PCA- y compararlos entre sí, se puede concretar el grado en que los daños estructurales y las variaciones en las rachas de viento afectan al comportamiento dinámico de la infraestructura.

2. MARCO TEÓRICO

2.1. Energía Eólica

Al igual que la mayoría de energías limpias, la eólica viene dada indirectamente por la radiación solar. La atmósfera recibe esta radiación de manera irregular y heterogénea, por lo que unas zonas del planeta sufren mayor calentamiento que otras, lo cual deriva en gradientes de presión, ya que el aire caliente tiende a subir, mientras que el aire frío es más pesado. Estos gradientes de presión se traducen en movimiento de grandes masas de aire atmosférico, **el viento**.

Estas corrientes de aire conllevan un gran valor de energía cinética, debido a la velocidad del viento y a la gran cantidad de kilogramos de aire que se desplaza. Con el avance en la investigación de energías renovables, el ser humano ha sido capaz de aprovechar parte de esa energía cinética que tienen las corrientes de aire mediante **parques eólicos**, convirtiéndola en energía eléctrica o mecánica.



Estos parques, surgidos a comienzos de la década de 1980, están compuestos por un gran número de molinos que llevan a cabo la conversión de energía cinética a energía eléctrica. El viento hace que las aspas -situadas en la parte superior de la torre- se muevan de forma circular, que a su vez están unidas a un eje común.

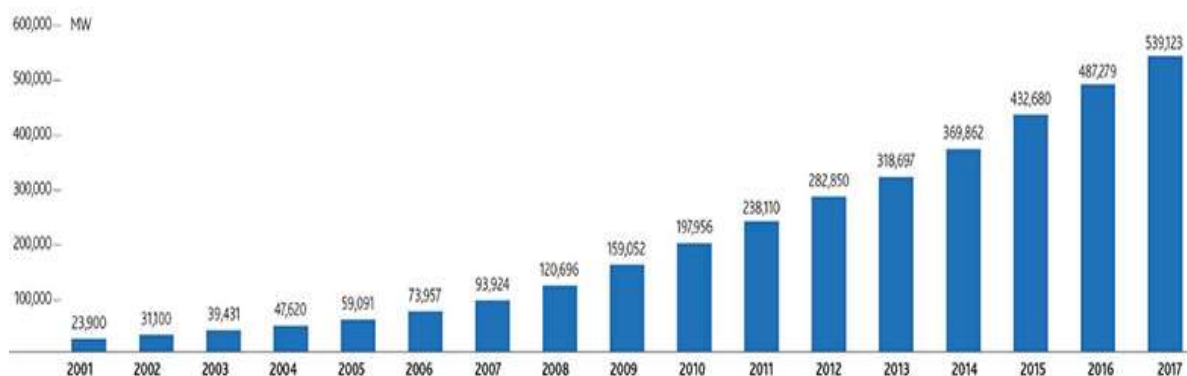
Antiguamente, gracias al giro de las aspas, se obtenía energía mecánica, frecuentemente para moler grano. Con el avance de la tecnología, este giro del eje se ha sabido aprovechar para la generación de electricidad.

Mediante un **alternador**, instalado alrededor del eje giratorio, es posible convertir la energía cinética acumulada en el eje en energía eléctrica. Al someter un elemento conductor a un campo magnético variable, se genera una fuerza electromotriz, cuyo valor viene dado por la Ley de Faraday, y se traduce en electricidad de **corriente alterna**.

2.2. Impacto actual

La gran demanda energética actual, sumado a los grandes avances tecnológicos, han permitido que la energía eólica haya crecido de manera exponencial los últimos años. Gracias a estos avances los parques eólicos cada vez son más fáciles de producir, y el aprovechamiento de la energía cinética del viento se ha optimizado hasta niveles muy altos.

En el año 2014 la energía eólica representaba el 5% de la producción energética mundial. Hoy en día, este porcentaje es casi el doble, un 9%. Dinamarca es el país que más contribuye a este desarrollo, con un 25% de energía producida mediante parques eólicos. Otros países como China, EEUU, Alemania, India y **España** también se encuentran en vanguardia, con alrededor del 20% de su producción total extraída del viento. Siendo en España la segunda fuente de energía para el sistema eléctrico, detrás de la nuclear, aportando **47.896 GWh** a la red.



Potencia instalada acumulada (2001-2017)

Dado que su implantación cada vez es más asequible, esta fuente de energía resulta tremendamente útil en zonas aisladas a las que no es posible abastecer de manera convencional, como desiertos o zonas de mayor altitud.

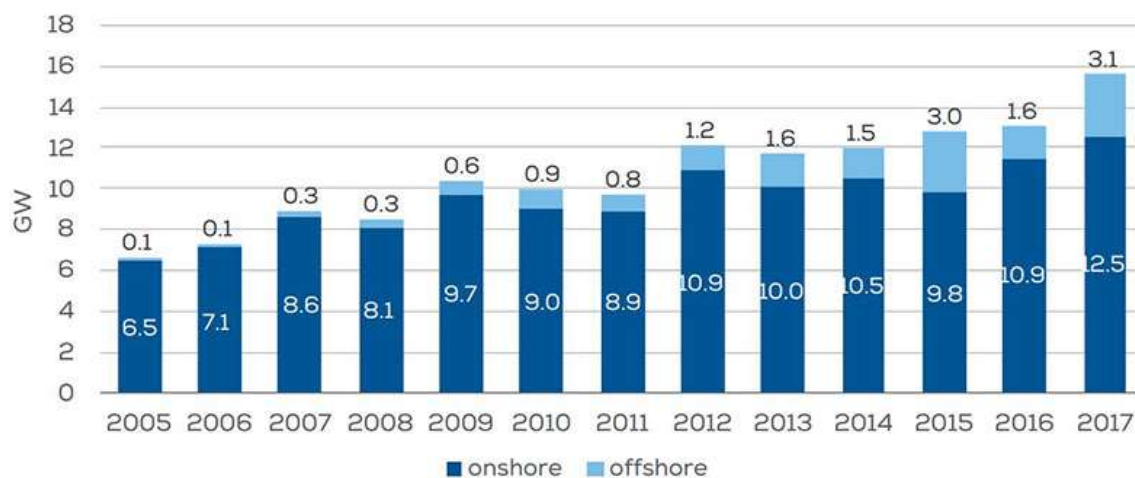
Quizá el mayor inconveniente a día de hoy es que resulta inviable abastecer un núcleo urbano únicamente con energía eólica, ya que la producción viene condicionada por las rachas de viento, cosa que no se puede controlar a nuestro antojo. Todavía no es posible almacenar grandes cantidades de energía eléctrica, por lo que el consumo de la energía eólica debe ser prácticamente instantáneo, lo que implica que en épocas sin grandes rachas de viento, la producción de esta energía decae inexorablemente.

2.3. Implantación al medio marino

Como venimos comentando, el auge de las energías renovables, en concreto la eólica, ha forzado al ser humano a buscar formas de mejorar este proceso, como una mejor localización de dichos parques.

Los **parques eólicos offshore** –por su nombre en inglés- tienen ciertas ventajas respecto a los situados en tierra firme, aunque también algunos inconvenientes. Se estima que la velocidad del viento en el mar es 1 m/s más rápida que en tierra, debido a la ausencia de obstáculos. Por el mismo motivo, las rachas de viento son más estables y uniformes cuando nos adentramos en el océano. Por último cabe resaltar que el impacto visual, sonoro y las limitaciones por superficie se reducen considerablemente al estar aproximadamente 20km mar a dentro.

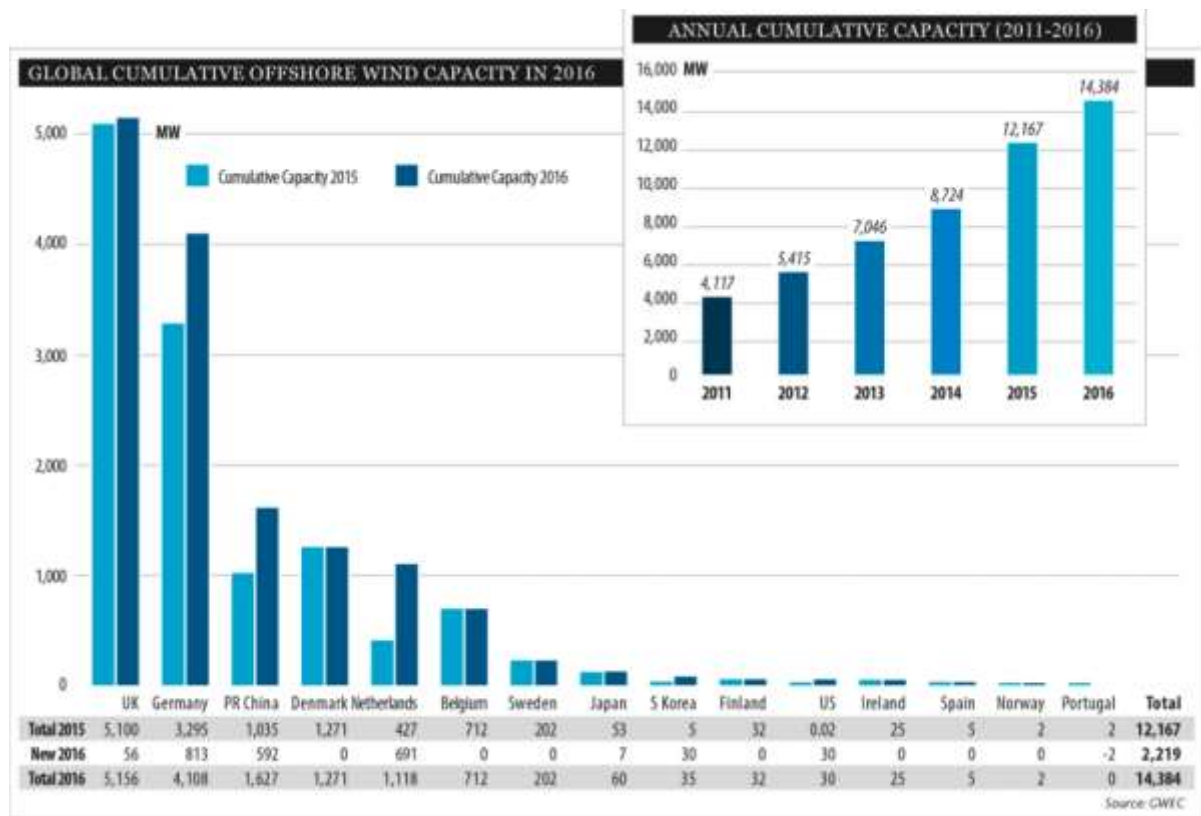
Aun así, estos parques suponen una pequeña parte sobre el total de energía eólica producida, básicamente por el gran coste que supone las infraestructuras necesarias, así como su posterior mantenimiento. Todo y que las torres de los parques offshore pueden ser de menor altura porque aprovechan mejor las rachas de viento, el coste que supone cimentar y construir un molino eólico en el océano es muy elevado respecto a uno en tierra.



Potencia eólica onshore/offshore instalada por año (2005-2017)

2.4. Parques offshore

Actualmente hay numerosos parques offshore operando, algunos incluso compiten en potencia con las instalaciones convencionales. El avance en logística permite que estos parques estén situados en zonas de mayor profundidad, aunque prácticamente sólo **Europa -88% del total-** saca partido de esta tecnología, con 81 proyectos instalados a finales de 2016. El 12% restante se lo reparten países como China, Japón, Corea del Sur y Estados Unidos.



Capacidad energética offshore por países (2016)

Reino Unido lidera el mercado eólico offshore, representando el 36% de la capacidad total en el mundo, seguido de Alemania con el 29%.

A continuación se muestra la tabla de los 10 parques eólicos más importantes del mundo:

Nombre del parque	Ubicación	Potencia [MW]
London Array	Essex, Reino Unido	630
Gemini Wind Farm	Groningen, Holanda	600
Gode Wind 1 y 2	Norderney, Alemania	582

Gwynt y Mor Park	Bahía de Liverpool, Reino Unido	576
Greater Gabbard Park	Suffolk, Reino Unido	504
Anholt Park	Dinamarca	400
BARD Offshore 1	Isla de Borkum, Alemania	400
Global Tech 1	ZEE, Alemania	400
West of Duddon Sands	Barrow-in-Furnes, Reino Unido	389
Walney Park	Mar de Irlanda, Reino Unido	367,2

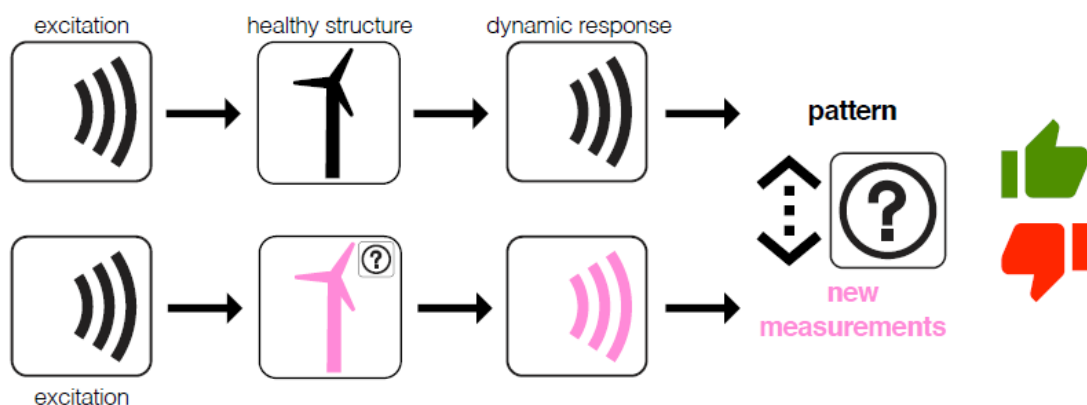
2.5. Detección de daños

Como no podría ser de otra forma, los parques eólicos offshore no son todo ventajas. Todo y que los parques offshore aprovechan mejor la energía del viento, no son la opción habitual a la hora de implantar nuevos parques. Las principales desventajas de este tipo de instalaciones vienen ligadas al factor económico, entre otras dificultades logísticas.

Tanto la construcción de los molinos eólicos, como el **mantenimiento** de los mismos, suponen un coste muy elevado para las compañías, ya que trabajar bajo el agua siempre es una dificultad añadida. En vista de ayudar a la evolución e implementación de este tipo de parques, la detección de daños en la estructura es crucial para llevar a cabo el mantenimiento –sobretudo preventivo- de una manera más sencilla.

El modelo de detección utilizado se basa en el **Análisis de componentes principales** –PCA por sus siglas en inglés-.

La idea principal se basa en que cuando alguna de las partes de la estructura está dañada, la respuesta dinámica de la torre ante las vibraciones que detecta el acelerómetro varía respecto a cuando la estructura es saludable. Se crea un patrón de estructura sana, que corresponde a los datos recabados por los acelerómetros cuando la estructura se encuentra en perfecto estado. A la hora de crear este modelo base, se hacen pruebas a diferentes amplitudes (rachas de viento variable), con el fin de tener más puntos de referencia cuando usemos el PCA.



Cuando la estructura sufra algún defecto, la señal de los acelerómetros será diferente a cuando teníamos la estructura sana, por lo que nos pondrá en aviso de que algo ha cambiado en cuanto al **comportamiento dinámico** de la estructura. Este aviso resulta de gran ayuda a la hora de realizar el mantenimiento correspondiente, ya que la otra forma de detectar un defecto es ver el daño in situ.

Esta detección in situ del fallo resulta difícil cuando la estructura a estudiar se encuentra sumergida, lo que conlleva un gran proceso logístico.

En definitiva, la utilidad de este sistema es avisar de que algo no funciona como debería, teniendo en cuenta la velocidad variable del viento.

3. HARDWARE

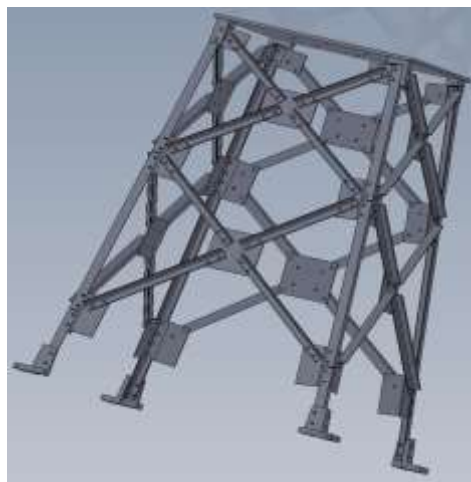
3.1. Torre prototipo

Nuestra torre representa una réplica a escala de una torre eólica offshore, apta para la toma de datos que se lleva a cabo. El objetivo de trabajar con este prototipo es recrear situaciones con las que se encontraría una torre instalada, en vista de poder replicar los resultados en una torre a tamaño real. Dicho prototipo consta de 2 partes principales, la parte superior, o "T", está formada por 4 barras gruesas de acero, sobre las que estará el excitador. La parte inferior, o jacket, es la parte de la estructura que quedaría sumergida en el océano. El jacket es la zona que se examina más a fondo, donde están 5 de los 8 acelerómetros, debido a que son los defectos en esta zona sumergida los que tratamos de detectar.



Parte superior/Jacket

La estructura está conformada por **barras y chapas**. Esta estructura piramidal consta de 32 barras de acero estructural S275JR de diferentes longitudes, chapas de acero DC01 LFR y otros elementos como la tornillería



Vista 3D de la parte inferior de la torre

Características de los aceros **S275JR** y **DC01**:

	<u>S275JR</u>	<u>DC01</u>
Contenido de Carbono: C [%]	<0,18	<0,1
Límite elástico: R_e [MPa]	225-275	140-280
Resistencia a tracción: R_m [MPa]	410-560	270-400
Alargamiento mínimo: A [%]	≥ 23	≥ 28

En lo referente a la sujeción entre chapas y barras, el total de tornillos que se usan para sujetar todo el conformado es de 224, además de tuercas –con sistema antiroscante- y arandelas. Con el fin de obtener unos datos lo más acertados posibles, se roscan todos los tornillos con un par de **12Nm**, asegurando que los puntos de unión no pierden fuerza.

Los tornillos utilizados son de acero, con cabeza hexagonal y M5 de diámetro. Por otro lado, las tuercas, también hexagonales, disponen de un sistema que bloquea el tornillo una vez roscado, impidiendo que los factores externos debiliten la unión.

3.2. Acelerómetros

Los acelerómetros, que harán las veces de sensores, son una pieza clave a la hora de detectar alguna anomalía en el comportamiento dinámico de la estructura. En nuestro caso particular, disponemos de 8 acelerómetros triaxiales **PIEZOTRONIC modelo 356A17**.

Este dispositivo en concreto dispone de una sensibilidad fija en el voltaje, sin importar el tipo de cable utilizado ni su longitud. Su señal de salida es de baja impedancia, por lo que puede transmitir a través de largos cables en ambientes hostiles sin perder calidad de señal. Dispone también de un Self-test intrínseco, con el que monitorizar el funcionamiento del dispositivo, ayudando a solucionar ciertos fallos que puedan aparecer.

Este dispositivo viene acompañado por un cable que se trifurca, dando una salida para cada componente X, Y y Z. Esta 3 salidas se conectan al cartucho que procesa por separado lo percibido por cada una de las componentes cartesianas.



Acelerómetro PIEZOTRONIC 356A17.



Salidas X, Y y Z del acelerómetro

Este acelerómetro digital dispone de las siguientes **prestaciones de funcionamiento**:

- Sensibilidad: 51 mV/(m/s²)
- Rango de medida: ±90 m/s² pk

- Rango de frecuencia ($\pm 5\%$): 0,5 – 3000 Hz
- Rango de frecuencia ($\pm 10\%$): 0,4 – 4000Hz
- Frecuencia de resonancia: ≥ 14 kHz
- Resolución (1 – 10000 Hz): 0,0006 m/s² rms
- Respuesta de fase ($\pm 5^\circ$): 2 – 4000 Hz
- Desalineamiento: $\leq 1\%$

Además de las **limitaciones físicas**:

- Límite de sobrecarga (choque): ± 49000 m/s² pk
- Rango de temperatura (funcionamiento): de -54 a 80°C

3.3. Devices NI

Disponemos de 6 dispositivos de National Instruments –modelo **NI 9234**– a los que conectamos nuestros acelerómetros. A su vez, estos cartuchos están insertados en un chasis también de National Instruments –modelo **cDAQ-9188**–. Gracias a estos dispositivos, podemos procesar las señales que reciben los acelerómetros cuando el excitador está en funcionamiento, pudiéndolas convertir al formato con el que trabajaremos.

El **cDAQ-9188** es un chasis Ethernet CompactDAQ, que consta de 8 ranuras de entrada –solo utilizaremos 6–, está diseñado para sistemas pequeños, remotos y distribuidos de medidas de sensor. El chasis controla la temporización, la sincronización y la transferencia de datos entre los módulos de E/S de la Serie C y un servidor externo. Puede usar este chasis con una combinación de módulos para poder realizar diversas medidas de E/S analógica, E/S digital y contador/temporizador. El chasis también tiene cuatro contadores/temporizadores de 32 bits de uso general.



Chasis NI cDAQ-918

El **NI-9234** puede medir señales desde sensores piezoeléctricos electrónicos integrados (IEPE) y no IEPE como acelerómetros, tacómetros y sensores de proximidad. El NI-9234 también es compatible con sensores TEDS inteligentes. El NI-9234 proporciona un amplio rango dinámico e incorpora acoplamiento de AC/DC y acondicionamiento de señales IEPE seleccionable por software. El canal de entrada mide las señales simultáneamente. Cada canal también tiene filtros ‘anti-aliasing’ integrados que ajustan automáticamente su velocidad de muestreo. Cuando es usado con software de NI, este módulo ofrece funcionalidad de procesamiento para monitoreo de condición como análisis de frecuencia y rastreo de orden.



Cartucho NI 9234

3.4. Excitador y amplificador de potencia

Por razones obvias de espacio y logística, no podemos experimentar con una torre completa, así que en el lugar donde estarían situadas las aspas del molino se instala un excitador inercial. Cuando se encuentre en funcionamiento, este excitador transmitirá las vibraciones causadas por la energía cinética del viento, a su vez simulará peso de esa pieza de la estructura.



Excitador, situado en la parte superior

El excitador se conecta al amplificador de potencia mediante un cable específico. Este amplificador intensificará la señal de ruido que previamente introducimos en el generador de funciones. El amplificador dispone de un regulador de potencia, que mantendremos fijo, ya que no está escalado y sería imposible ser precisos en esa medida.

3.5. Generador de funciones

Concretamente, nuestro modelo es el **GW INSTEK AF-2005**. Este generador de señales basado en DDS (síntesis digital directa) está diseñado para satisfacer los requisitos educativos e industriales básicos como una fuente de señales precisa y asequible que cubre formas de onda de salida sinusoidal, cuadrada (pulso), rampa (triángulo), ruido y arbitraria.



Durante nuestros ensayos, la forma utilizada será la de ruido, ya que genera unas vibraciones que se aproximan más a las que podrían producir unas rachas de viento sobre la torre. Cuando seleccionamos la opción de ruido, no nos permite modificar la frecuencia, que se mantiene constante sin importar la variación de amplitud.

La tasa de muestro es de 20MSa/s, con una resolución vertical de 10 bits y memoria de 4.000 puntos. Todo esto proporciona al usuario un entorno flexible para la creación de la salida de forma de onda específica que se necesita. La resolución de 0,1Hz de formas de onda sinusoidal, cuadrada y triangular y el ciclo de servicio es ajustable de 1% ~ 99%.

El aparato dispone de una pantalla de 3,5" en la que se muestra -a tiempo real- la amplitud y desviación DC, además de información paramétrica útil para el usuario.

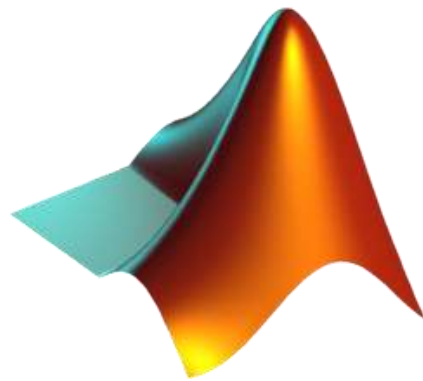
4. SOFTWARE

4.1. NI-Max

Este programa, que pertenece a National Instruments, transcribe las señales del chasis de cartuchos, compatibilizándolos con nuestro sistema. Esta es la única función del programa NI-Max, ya que una vez sincronizado con Matlab, obtendremos los datos directamente en su interface.

4.2. Matlab 2017.b

Esta herramienta de software matemático, creada por Cleve Moler, que nace el año 1984 ha sido, y sigue siendo desarrollada por el propio Cleve Moler con la ayuda de **MathWorks**. El nombre de MATLAB es un acrónimo procedente de los términos anglosajones MATrix LABoratori – laboratorio de matrices-.



Disponemos de la última versión de este software, el Matlab 2017.b -20 de septiembre de 2017-, el cual es compatible con Unix, Windows, Mac OSX y Linux. Ofrece un entorno de desarrollo integrado, con lenguaje propio a la hora de programar –lenguaje M-. En el ámbito más básico, Matlab otorga al usuario la posibilidad de manipular matrices, crear representaciones gráficas de datos y funciones, entre otras muchas. Concretamente, la creación de algoritmos es una de la herramientas que utilizamos en este software, como puede verse en el ANEXO 2, para el algoritmo creado para recabar datos.

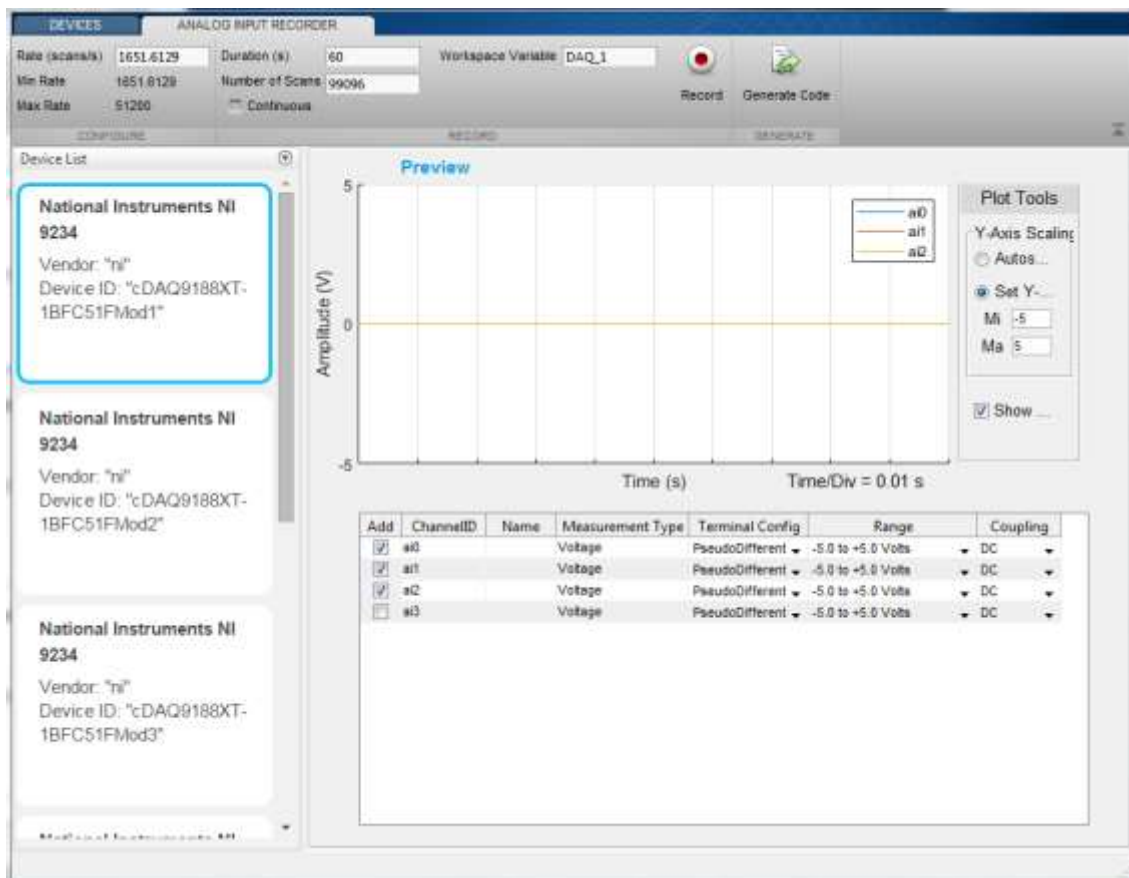
No es de extrañar que millones de ingenieros y científicos utilicen esta plataforma a la hora de analizar y diseñar sistemas y productos. Está presente, por ejemplo, en sistemas de seguridad activa de automóviles, naves espaciales, monitorización sanitaria, entre otros muchos campos.

Por otro lado, también nos ofrece la oportunidad de comunicarnos con otros programas, como el NI-Max, e incluso con otros tipos de hardware, con el objetivo de trabajar en un único interface. Este ha sido uno de los motivos para llevar a cabo el proyecto con esta plataforma, más no el único. Podemos trabajar en un entorno interactivo e intuitivo que dispone de un lenguaje de alto nivel para realizar cálculos, además de gráficas como las que vemos en el ANEXO 3.

Dentro de Matlab podemos descargar diferentes herramientas –o Toolboxes- complementarias con el fin de ampliar el espectro de posibilidades de trabajo.

4.2.1. Analog Input Recorder

Concretamente usamos el Toolbox ‘**Analog Input Recorder**’, que nos ayuda a crear un código base, sobre el que desarrollar el mostrado en el ANEXO 2, que utilizaremos para registrar las señales enviadas por los sensores.



Interface de ‘Analog Input Recorder’

Con esta herramienta seleccionamos el cartucho NI que analizará durante un tiempo preseleccionado –parte superior-. Del mismo modo, podemos modificar la frecuencia a la hora de recopilar datos, el cual dejaremos en el mínimo posible, 1651,6129 escáner por segundo, que durante nuestros 60 segundos son 99097 escaneos.

Cuando pulsamos ‘Record’, el Analog Input Recorder comienza su obtención de datos, todo y que con esta aplicación sólo podemos recabar datos de un cartucho al mismo tiempo. Cuando termina el proceso, usamos la opción ‘Generate Code’, que nos da un código base, que modificamos para poder obtener lo percibido por los 8

acelerómetros, a la vez que incorporamos comandos para guardar la sesión, representar gráficamente los resultados, renombrar variables, etc.

5. OBTENCION DE DATOS

5.1. Ensayos estructura sana (30 iteraciones)

Una vez llevado a cabo todo el proceso de montaje explicado en el ANEXO 1, procedemos a realizar los ensayos en el prototipo.

En primer lugar, realizaremos 30 ensayos -60 segundos de duración cada uno- con la estructura completamente saludable. Lo recabado en estas 30 iteraciones será la base a la que se atenga el PCA para intentar distinguir cuando la torre sufre algún defecto.

Dividimos las 30 iteraciones en tres amplitudes distintas -2, 4 y 6 Vpp- que simularán las condiciones que sufriríamos en un caso real, con rachas de viento variable. Para iniciar el ensayo, introducimos en el generador de funciones la amplitud de ruido correspondiente. Acto seguido pulsamos la tecla 'OUTPUT', que pone en marcha el excitador. Tras unos 5 segundos, la señal del excitador se estabiliza, y es cuando corremos nuestro programa de recopilación de datos de Matlab -mostrado en el ANEXO 3-. El proceso en los ensayos será igual para todas las iteraciones. Todas ellas se guardarán la misma carpeta con una nomenclatura específica, vital en el primer paso del volcado de datos.

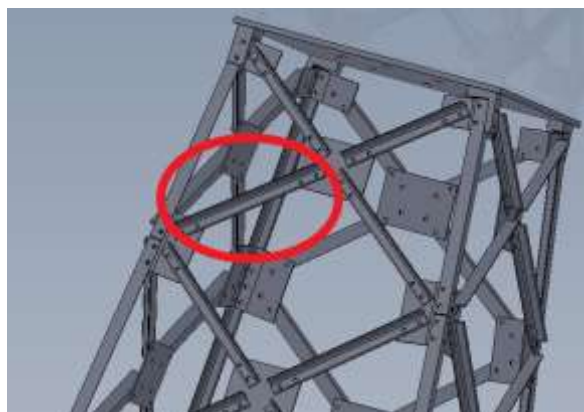
X_Y_ZA.mat

X: -Del 1 al 4- Ensayos estructura original, recambio saludable, corte de 5mm y falta de tornillo.

Y: -Del 1 al 5/10- Número de la iteración.

Z: -2, 4 o 6- Amplitud de la señal de ruido.

En la imagen se muestra el lugar en el que se sitúa la barra que manipularemos durante los diferentes ensayos:



Ubicación de la barra modificable

5.2. Ensayos sustituyendo una barra por otra igualmente sana (15 iteraciones)

En estas 15 nuevas iteraciones -5 a cada amplitud- reemplazamos la barra señalada anteriormente por otra completamente idéntica. Los datos recopilados serán un indicativo de que la simple sustitución de una barra sana por otra no causa ningún tipo de dispersión en los datos, reforzando así el modelo base que servirá de patrón.



Barra saludable sustituta

5.3. Ensayos sustituyendo por una barra seccionada 5mm (15 iteraciones)

En este tramo, volvemos a desmontar la réplica saludable que habíamos montado en el apartado anterior para colocar una tercera barra, esta con un corte de 5mm de profundidad y realizado en 1/3 de su longitud. Se ensaya con esta barra defectuosa a las 3 amplitudes de ruido, obteniendo así los primeros datos con un fallo en la estructura, que al procesarlos con el PCA, deberían alertar que la



estructura
no se
encuentra
totalmente
sana.

$$d = 5 \text{ mm} \quad X = L/3$$

Barra seccionada

5.4. Ensayos quitando un tornillo (15 iteraciones)

En último lugar, volvemos a colocar la barra original en la estructura, pero esta vez no usamos todos los tornillos, dejando un agujero sin sujeción. Este segundo defecto, al igual que el corte en la barra, debería alterar el comportamiento dinámico de la torre ante las vibraciones inducidas. Se retira el tornillo más próximo al acelerómetro más cercano, en vista de intentar resaltar los valores defectuosos.

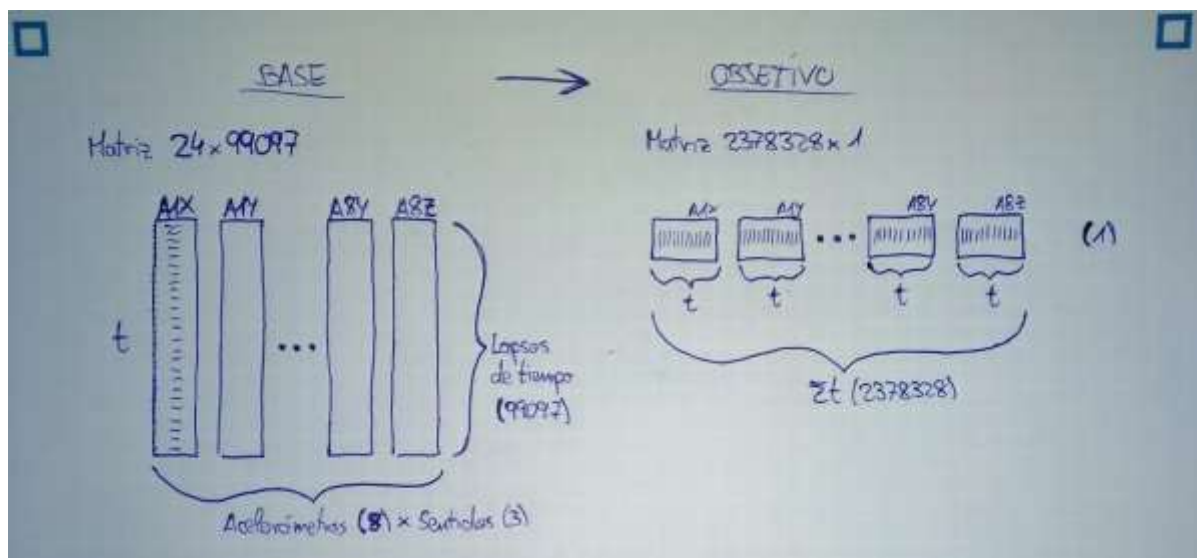


Barra sin uno de los tornillos

6. PROCESAMIENTO DE DATOS (PCA)

Una vez concluida la fase experimental es hora de compilar los datos obtenidos mediante nuestro sistema de procesado. Obteniendo datos de hasta 75 iteraciones diferentes, 30 estructura original, 15 recambio sano, 15 seccionado de 5mm y 15 con la falta de un tornillo. Cada una de estas iteraciones contiene una matriz de 24 columnas -8 acelerómetros x 3 direcciones- y 99097 filas -1651,6129 scans/s x 60 segundos-.

El problema surge porque los datos que se introducen en el PCA tienen que estar todos recogidos linealmente, en una sola fila. Esta única fila constará de todos los datos de todos los acelerómetros, ordenados como se muestra en la nota:



Nota explicativa conversión de las matrices

Esta transformación de matrices bien podría hacerse de forma manual, copiando cada columna detrás de la anterior, pero conllevaría un trabajo largo y tedioso. Con el fin de evitar esta carga de trabajo y con la ayuda de Matlab podemos automatizar todo el proceso mediante un código de programación. Este código cargará uno a uno y ordenadamente los 75 archivos **X_Y_ZA.mat**, y usando el comando 'reshape' dentro de un bucle hacemos que reordene las matrices 24x99097 en la otras del tipo 2378328x1.

6.1. Volcado de datos

```

1 - for vol=2:2:6
2 -     sensors = 8*3;
3 -     time = 99097;
4 -     X = zeros(10,sensors*time);
5 -     for i=1:10
6 -         var=strcat('1_',num2str(i));
7 -         var=strcat(var,'_A');
8 -         var=strcat(var,num2str(vol));
9 -         var=strcat(var,'.mat');
10 -        load(var);
11 -        X(i,:) = reshape(data,1,[]);
12 -    end
13 -
14 -    Y = zeros(15,sensors*time);
15 -    for j=2:4
16 -        for i=1:5
17 -            var=strcat(num2str(j));
18 -            var=strcat(var,'_');
19 -            var=strcat(var,num2str(i));
20 -            var=strcat(var,'_A');
21 -            var=strcat(var,num2str(vol));
22 -            var=strcat(var,'.mat');
23 -            load(var);
24 -            Y(i+5*(j-2),:) = reshape(data,1,[]);
25 -        end
26 -    end
27 -
28 -    varX=strcat('matrix_X_',num2str(vol));
29 -    varX=strcat(varX,'.mat');
30 -    varY=strcat('matrix_Y_',num2str(vol));
31 -    varY=strcat(varY,'.mat');
32 -    save(varX,'X')
33 -    save(varY,'Y')
34 - end

```

Tras procesar todos nuestros .mat por el código descrito arriba, disponemos de un total de 6 matrices, que sustituyen la gran cantidad que teníamos antes. Están las 3 matrices Xs, de proporciones 2378382x10, que representa los valores de los 8 acelerómetros en las 3 direcciones durante los 60 segundos del ensayo -2378382-, y con 10 filas que representan el número de repeticiones estipulado -10 iteraciones-.

Por otro lado, las 3 matrices Ys, de proporciones 2378382x15, representan lo mismo que las Xs, pero esta consta de 15 filas, ya que se hacen 15 iteraciones con la torre modificada.

Llegados a este punto, tenemos los datos en la forma necesaria para crear el modelo mediante el PCA –Análisis de las componentes principales-.

6.2. Código PCA

La principal función del PCA es crear una nube de puntos con los datos obtenidos de la estructura saludable, creando así un patrón para cuando la estructura tiene un comportamiento dinámico adecuado. A partir de ahí, se procesan de la misma forma los datos con defectos en la estructura, que deberían dar una nube de puntos distinta a la original.

Gracias a este código, convertimos las matrices Xs e Ys en una representación gráfica.

- Sección 1

```

1 - clf % Clear current figure window
2 - close all
3 - clear all
4
5 %% Variables
6
7 - N = 24; % "Sensores" (8 sensores y 3 direcciones por sensor)
8 - rL = 99097; % Instantes de tiempo
9 - mostra = 1; % Sanos (D0)
10 - nfallos = 2; % Daños (D1..D2)
11 - nX = 10; % Tamaño de cada caso (D0) para el modelo PCA, train
12 - nY = 5; % Tamaño de cada caso (D0..D2) para el test
13 - filasY = (mostra+nfallos)*nY; % Núm. de filas Y (test) a introducir: 15
14
15 %% Datos
16
17 % X e Y:
18
19 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20 %
21 % load ('matrix_X_2.mat'); % TRAIN, tiene nX D0's
22 % load ('matrix_Y_2.mat'); % TEST, tiene nY D0's .. nY D2's
23 % load ('matrix_X_4.mat'); % TRAIN, tiene nX D0's
24 % load ('matrix_Y_4.mat'); % TEST, tiene nY D0's .. nY D2's
25 - load ('matrix_X_6.mat'); % TRAIN, tiene nX D0's
26 - load ('matrix_Y_6.mat'); % TEST, tiene nY D0's .. nY D2's
27 %
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29

```

Después de cerrar y limpiar todo lo que estuviera utilizando Matlab, se introducen las variables fijas en el código, dándoles un nombre específico.

A continuación cargamos los datos obtenidos experimentalmente, nuestras 6 matrices, Xs e Ys.

- Sección 2:

```

30 %% PCA
31
32 %% MCGS:
33
34 mat = [X;Y]; % X + Y = 10 + 15 = 25 filas
35
36 dt = zeros(1,N); % Las N desviaciones estándar, una por "sensor" (solo datos X)
37 medias = zeros(1,N*rL); % La media de cada columna de X (solo datos X)
38 XT = zeros((mostra)*nX+filasY,N*rL); % X e Y
39
40 for j = 1:N
41     disp(strcat('Sensor',int2str(j)));
42     % We just compute the standard deviation of the X ((mostra)*nX rows)
43     % Standard deviation of all the healthy measures of "sensor" i=1:N
44     dt(j) = std(reshape(X(:,(j-1)*rL+1:j*rL),1,(mostra)*nX*rL));
45     for m = 1:rL % number of columns per block
46         % XT is the scaled matrix after the group-scaling
47         % princomp centers X by subtracting off column means
48         medias((j-1)*rL+m) = mean(X(:,(j-1)*rL+m));
49         XT(:,(j-1)*rL+m) = (mat(:,(j-1)*rL+m)-medias((j-1)*rL+m))/dt(j);
50     end
51 end
52
53 %% PCA:
54
55 [coeff,score,latent] = pca(XT(1:(mostra)*nX,:)); % (solo datos X)
56
57 % Se proyecta toda XT en el nuevo plano:
58
59 T = XT*coeff;
60

```

En esta sección se procesan las matrices con el sistema PCA, obteniendo la ordenación deseada para nuestros datos.

- Sección 3:

```

61 %% Gráficos
62
63 labels = [0*ones(1,nX)'; % X
64           1*ones(1,nY)'; % Y
65           2*ones(1,nY)';
66           3*ones(1,nY)'];
67
68 labels2 = 0*ones(1,nX)'; % X
69
70 simbolos = ["ko", "gd", "bs"];
71
72 % Versión 1:
73 figure(1)
74 gscatter(T(:,1),T(:,2),labels) % X + Y
75
76 % Versión 2:
77 figure(2)
78 gscatter(T(1:(mostra)*nX,1),T(1:(mostra)*nX,2),labels2) % X
79 hold on
80 for i = 0:nfallos
81     plot(T((mostra)*nX+1+(i*nY):(mostra)*nX+(i+1)*nY,1),...
82          T((mostra)*nX+1+(i*nY):(mostra)*nX+(i+1)*nY,2),simbolos(i+1)) % Y
83 end
84 legend('X-D0', 'Y-D0', 'Y-D1', 'Y-D2')
85 hold off
86 set.figure(2),'WindowState','maximized');
87
88 %%%%%%%%%%%
89 %
90 % print('Plots\PCA_2','-depsc')
91 % print('Plots\PCA_4','-depsc')
92 print('Plots\PCA_6','-depsc')
93 %
94 %%%%%%%%%%%
95
96 set.figure(2),'WindowState','normal');
97

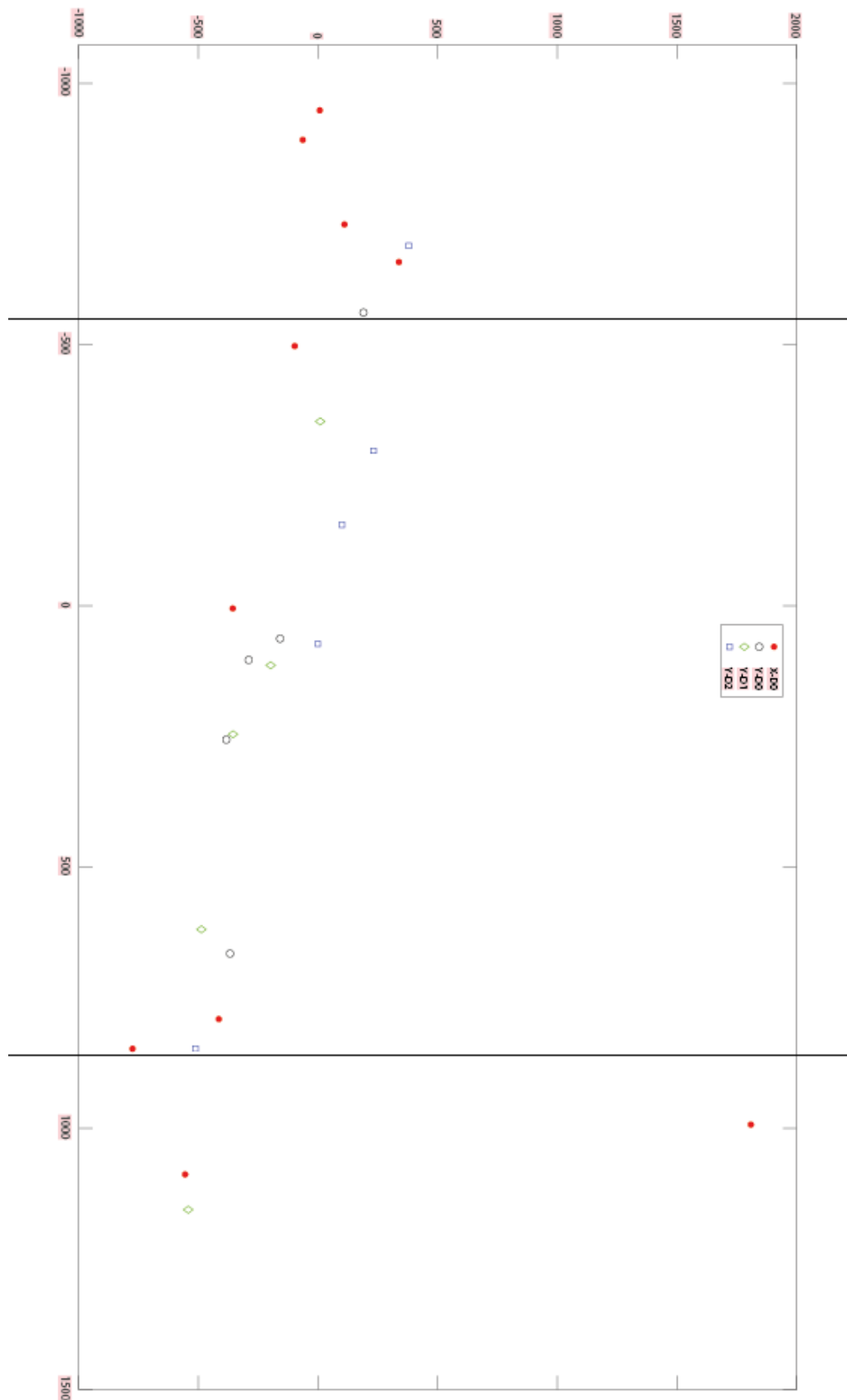
```

Por último, de manera similar al programa que usamos para recabar datos, el final de este código nos ilustra, en forma de gráfico, la dispersión de puntos que buscábamos.

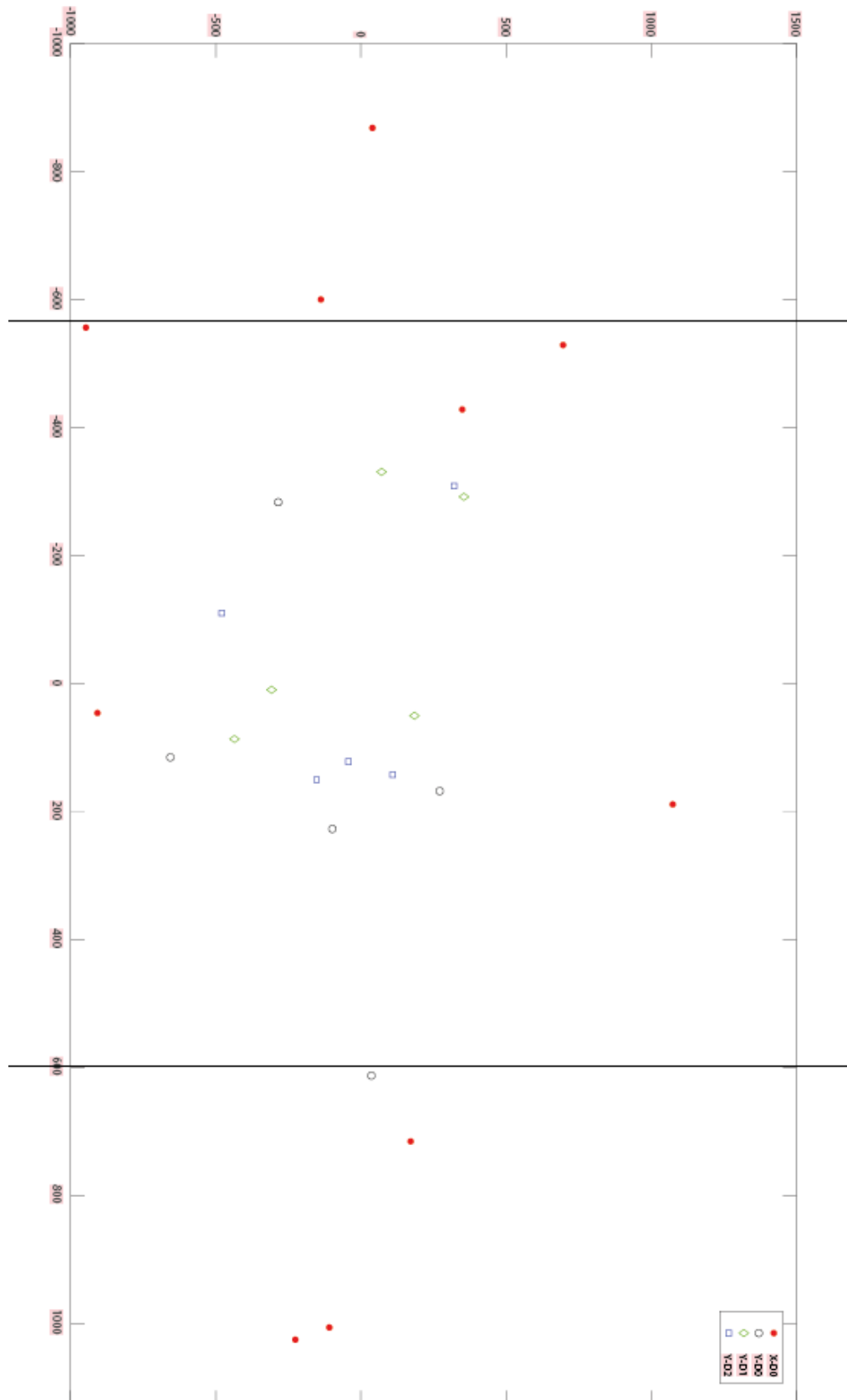
6.3. Nubes de puntos

Finalmente obtenemos los gráficos deseados, que se muestran en las siguientes imágenes, separadas por amplitud de ruido inducido.

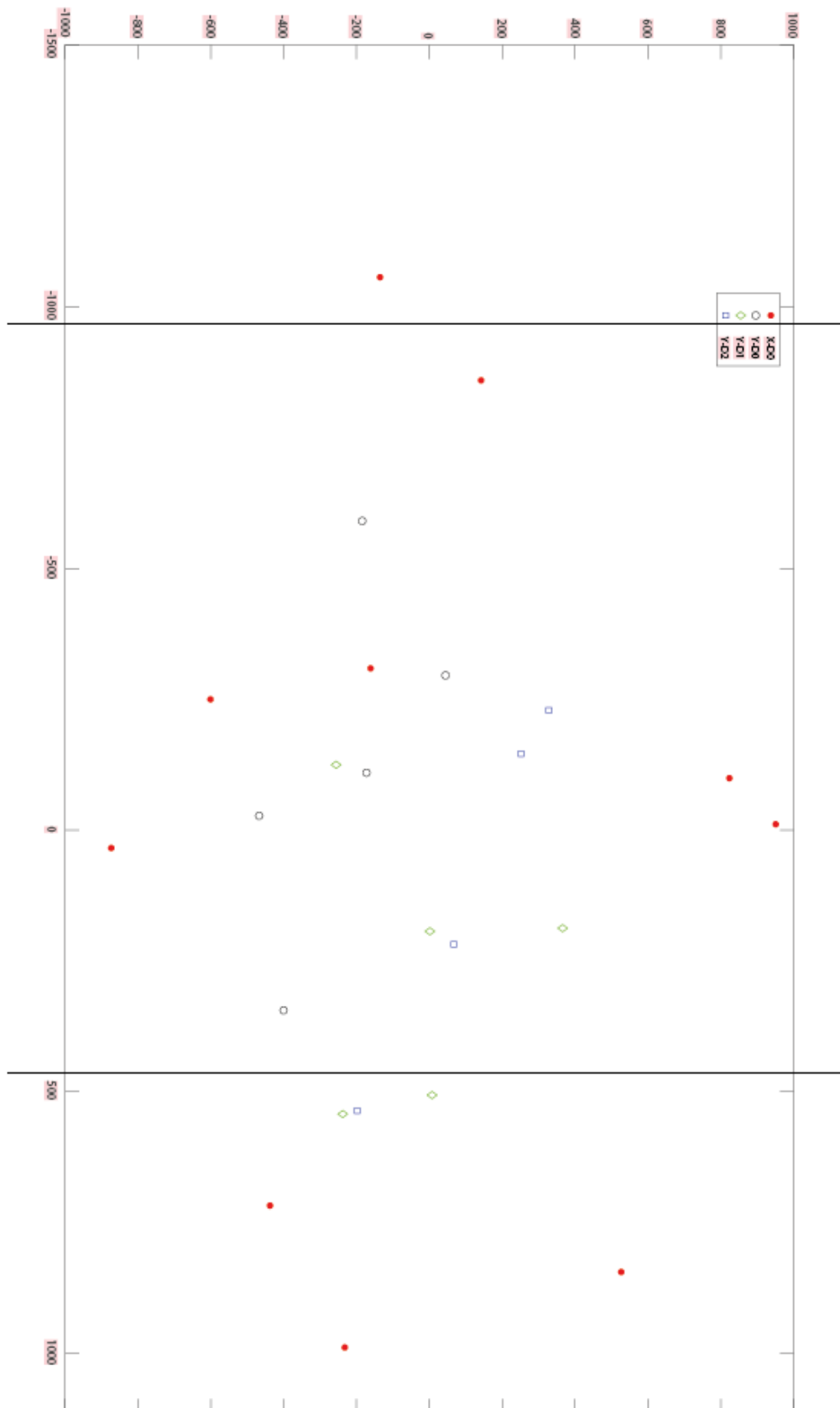
- Amplitud de ruido: **2Vpp**



- Amplitud de ruido: **4Vpp**



- Amplitud de ruido: **6Vpp**



6.4. Discusión de los resultados

Cabe decir que los test realizados hasta ahora han sido pruebas experimentales para comprobar el funcionamiento del equipo que hemos montado. Queda en un segundo plano la coherencia o utilidad de estos datos, ya que el verdadero objetivo era crear un prototipo apto para la toma de datos ante las vibraciones.

Una vez creado el modelo de PCA, representado en el apartado anterior como una dispersión de puntos, se observa que no hay una agrupación clara a simple vista. Visualmente los puntos obtenidos no parecen tener mucho significado, al menos en estas gráficas, ya que no existen agrupaciones distinguidas por amplitud o por la aparición de algún defecto.

Aunque visualmente no podamos interpretar los resultados, si hay una manera de indagar más en estos resultados. Mediante el contraste de hipótesis -basado en la media aritmética y la desviación típica de los puntos hallados- es posible que estos resultados adquieran algún sentido útil.

CONCLUSIONES

Tanto el montaje del Hardware –torre, acelerómetros, excitador y generador de funciones-, como el del Software –NI-Max y Matlab- han resultado un éxito. Gracias a la puesta a punto del equipo, hemos sido capaces de recabar datos a tiempo real del comportamiento dinámico de la estructura.

Una vez verificado que el equipo es operativo, se procede a la realización de pruebas para la detección de fallos. Al margen de la distinción entre estructura sana y defectuosa, la metodología utilizada en el proceso es completamente válida.

Como continuación de este proyecto, y utilizando los medios descritos a lo largo de la memoria, sería posible diferenciar el comportamiento cuando la estructura se encuentra saludable de cuando sufre algún defecto.



BIBLIOGRAFÍA/WEBGRAFÍA

Datos energía solar:

<https://upcommons.upc.edu/bitstream/handle/2117/91070/TFG.pdf>

<https://www.aeeolica.org/es/sobre-la-eolica/la-eolica-en-el-mundo/>

<http://www.cubasolar.cu/biblioteca/energia/Energia35/HTML/articulo05.htm>

Luo, N., Vidal, Y., & Acho, L. (Eds.). (2014). *Wind turbine control and monitoring*. Springer International Publishing.

Parques eólicos:

<https://elperiodicodelaenergia.com/los-10-mayores-parques-eolicos-marinos-del-mundo/>

Sistema de detección de daños:

Pozo, F.; Vidal, Y. Wind Turbine Fault Detection through Principal Component Analysis and Statistical Hypothesis Testing. *Energies* **2016**, 9, 3.

Propiedades metal torre:

http://www.constructalia.com/repository/transfer/es/resources/Contenido/02701331Foto_Big.pdf

http://automotive.arcelormittal.com/saturnus/sheets/K_ES.pdf

Acelerómetros:

<http://www.pcb.com/Products/model/356A17>

National Instruments:

<http://www.ni.com/es-es/support/model.cdaq-9188.html>

<http://www.ni.com/es-es/support/model.ni-9234.html>

Generador de funciones:

http://es.farnell.com/gw-instek/afg-2005/generador-de-funci-n-arb-dds-5mhz/dp/2100020?ost=AFG+2005&ICID=redirect-Y&CMP=os-geobanner-google&DM_PersistentCookieCreated=true

Matlab:

https://es.mathworks.com/help/matlab/learn_matlab/product-description.html



ANEXOS

ANEXO 1: Proceso de montaje del equipo

A1.1. Torre prototipo

Al comienzo del proyecto disponemos de la estructura prototipo montada por el equipo de mantenimiento, a excepción de los tornillos, que se sustituirán por otros y aseguraremos el mismo par en todos y cada uno de ellos.

Todos los tornillos y tuercas -224 en total- son sustituidos por otros con sistema antiroscante, para asegurarnos de mantener el mismo par de fuerza en todos los puntos de sujeción a lo largo del experimento. También se colocan arandelas entre el tornillo y la chapa.

Utilizamos una herramienta específica que nos permite saber cuándo el tornillo llega al par de fuerza deseado, en nuestro caso todos se hallan a **12Nm de par**.



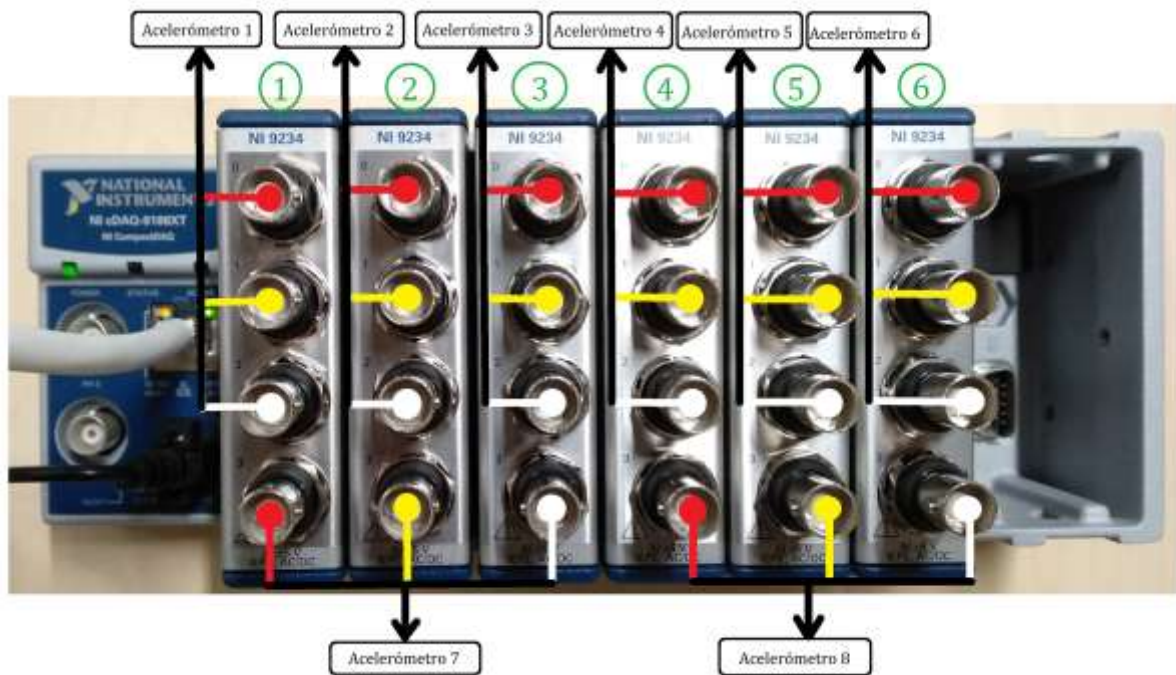
Llave de par ajustable

Esta tipo de llave, conocido como tipo click o de par ajustable, nos permite fijar un par en concreto y mantenerlo durante todo el ajuste de los tornillos de manera sencilla y precisa. El rango de precisión es de $\pm 4\%$ del valor del par ajustado.

A1.2. Acelerómetros y Devices NI

La instalación de los acelerómetros empieza por conectar los cables de salida X (rojo), Y (amarillo) y Z (blanco) en cada una de las entradas de los dispositivos de NI, que procesa los datos para compatibilizarlos con nuestro software. Cada uno de los 6 cartuchos dispone de 4 entradas –ai0, ai1, ai2 y ai3- a las que conectamos las salidas de los acelerómetros.

La disposición del cableado se muestra en este esquema:



Conexión de los acelerómetros a los cartuchos

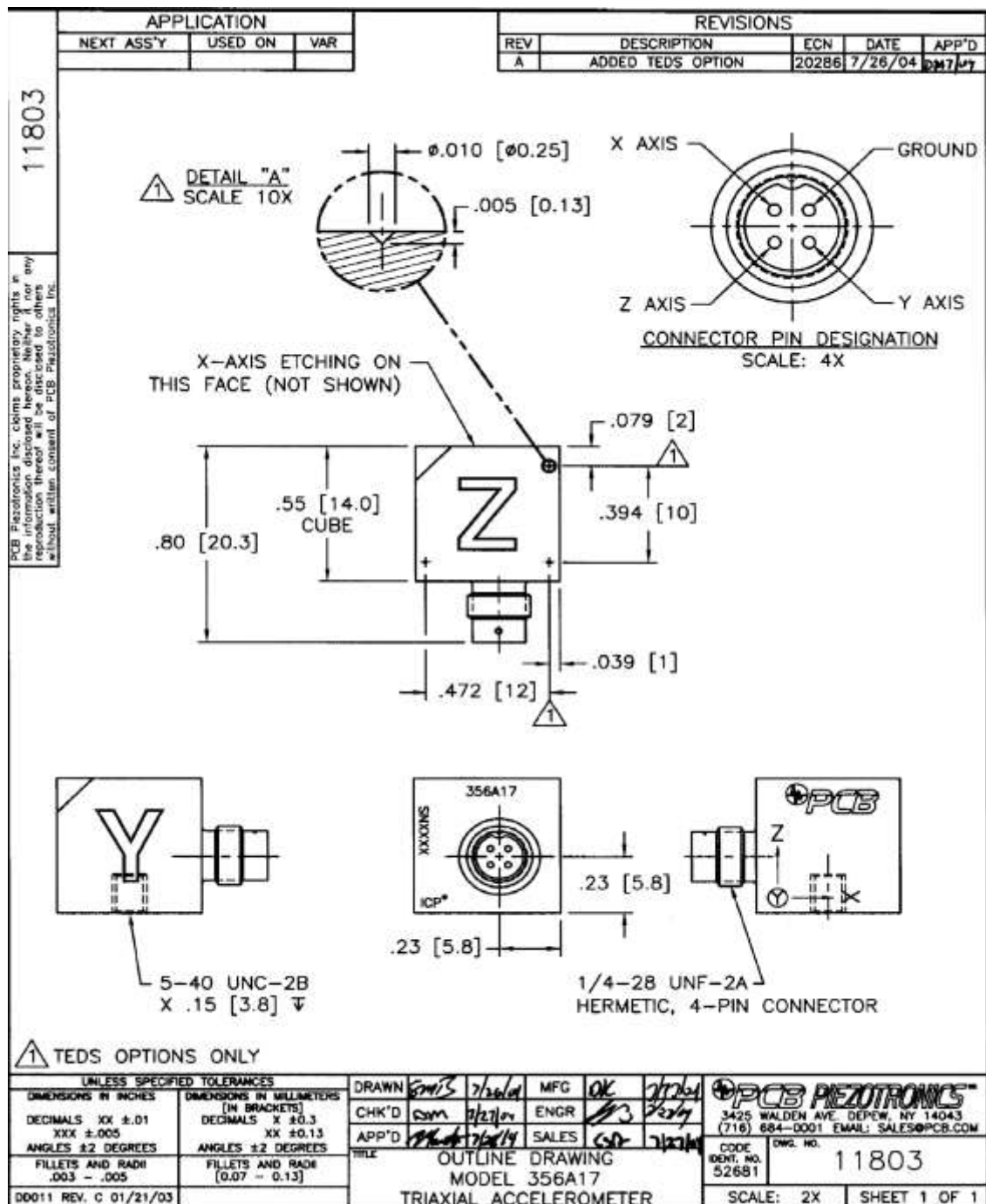
Al conectar el chasis de los cartuchos al ordenador, procedemos a realizar un Self-test con el que verificamos el funcionamiento del dispositivo desde nuestra interface.

En último lugar, los acelerómetros se disponen en los puntos predeterminados de la torre para los ensayos. De los 8 acelerómetros, 3 se sitúan en la parte superior del prototipo y los 5 restantes se distribuyen por el Jacket. Para fijar los acelerómetros en la estructura usamos cera de abeja, incluida en la caja del acelerómetro.



Chasis totalmente conectado

A continuación se muestra un plano detallado de los acelerómetros utilizados, así como de sus conexiones, sujeciones, acabados superficiales y demás:



Plano técnico PIEZOTRONIC 365A17

A1.3. Excitador, amplificador y generador de funciones

Con la ayuda del equipo de mantenimiento, se atornilla el excitador en lo alto de la torre, emplazado en el lugar que ocuparían las aspas. Este excitador consta de un único cable de salida, el cual se conecta al amplificador de señal. El amplificador se utiliza con el fin de transmitir y regular la señal que produciremos con el generador de señales.

Por tanto, el amplificador se conecta al generador de funciones mediante un cable BNC, específico para este instrumental.



Conexión amplificador-generador de funciones

ANEXO 2: Código Matlab

A2.1. Sección 1

1	%% Save data	
2		
3	save test.mat	
4		
5	%% Create Data Acquisition Session	
6	% Create a session for the specified vendor.	
7	s = daq.createSession('ni');	
8		
9	%% Set Session Properties	
10	% Set properties that are not using default values.	
11	s.DurationInSeconds = 60;	
12		

La línea 3 es un simple 'save' con el que poder ir registrando los ensayos realizados en un archivo .mat.

Posteriormente, la línea 7 crea una sesión que permite a Matlab recabar datos recibidos del chasis NI.

Por último, en la línea 11 especificamos el tiempo que durará la toma de datos cada vez que se haga correr este programa.

A2.2. Sección 2

13	%% Add Channels to Session	
14	% Add channels and set channel properties, if any.	
15		
16	%Primer acelerómetro	
17	channelA1X = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod1', 'ai0', 'Voltage');	
18	channelA1X.Name = 'A1X';	
19	channelA1Y = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod1', 'ai1', 'Voltage');	
20	channelA1Y.Name = 'A1Y';	
21	channelA1Z = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod1', 'ai2', 'Voltage');	
22	channelA1Z.Name = 'A1Z';	
23		
24	%Segundo acelerómetro	
25	channelA2X = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod2', 'ai0', 'Voltage');	
26	channelA2X.Name = 'A2X';	
27	channelA2Y = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod2', 'ai1', 'Voltage');	
28	channelA2Y.Name = 'A2Y';	
29	channelA2Z = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod2', 'ai2', 'Voltage');	
30	channelA2Z.Name = 'A2Z';	
31		
32	%Tercer acelerómetro	
33	channelA3X = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod3', 'ai0', 'Voltage');	
34	channelA3X.Name = 'A3X';	
35	channelA3Y = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod3', 'ai1', 'Voltage');	
36	channelA3Y.Name = 'A3Y';	
37	channelA3Z = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod3', 'ai2', 'Voltage');	
38	channelA3Z.Name = 'A3Z';	
39		
40	%Cuarto acelerómetro	
41	channelA4X = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod4', 'ai0', 'Voltage');	
42	channelA4X.Name = 'A4X';	
43	channelA4Y = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod4', 'ai1', 'Voltage');	
44	channelA4Y.Name = 'A4Y';	
45	channelA4Z = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod4', 'ai2', 'Voltage');	
46	channelA4Z.Name = 'A4Z';	
47		

```

48 %Quinto acelerómetro
49 - channelA5X = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod5', 'ai0', 'Voltage');
50 - channelA5X.Name = 'A5X';
51 - channelA5Y = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod5', 'ai1', 'Voltage');
52 - channelA5Y.Name = 'A5Y';
53 - channelA5Z = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod5', 'ai2', 'Voltage');
54 - channelA5Z.Name = 'A5Z';
55
56 %Sexto acelerómetro
57 - channelA6X = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod6', 'ai0', 'Voltage');
58 - channelA6X.Name = 'A6X';
59 - channelA6Y = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod6', 'ai1', 'Voltage');
60 - channelA6Y.Name = 'A6Y';
61 - channelA6Z = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod6', 'ai2', 'Voltage');
62 - channelA6Z.Name = 'A6Z';
63
64 %Séptimo acelerómetro
65 - channelA7X = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod1', 'ai3', 'Voltage');
66 - channelA7X.Name = 'A7X';
67 - channelA7Y = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod2', 'ai3', 'Voltage');
68 - channelA7Y.Name = 'A7Y';
69 - channelA7Z = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod3', 'ai3', 'Voltage');
70 - channelA7Z.Name = 'A7Z';
71
72 %Octavo acelerómetro
73 - channelA8X = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod4', 'ai3', 'Voltage');
74 - channelA8X.Name = 'A8X';
75 - channelA8Y = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod5', 'ai3', 'Voltage');
76 - channelA8Y.Name = 'A8Y';
77 - channelA8Z = addAnalogInputChannel(s, 'cDAQ9188XT-1BFC51FMod6', 'ai3', 'Voltage');
78 - channelA8Z.Name = 'A8Z';
79

```

En esta segunda sección –desde la línea 17 hasta la 78- añadimos a la sesión y damos nombre a todos los sensores utilizados, cada acelerómetro consta de 3 componentes X, Y y Z, conectadas a las diferentes entradas de los cartuchos NI. Por ejemplo; el Primer acelerómetro tiene sus 3 salidas conectadas en las entradas ‘a0’, ‘a1’ y ‘a2’ del cartucho ‘Mod1’; mientras, el Séptimo acelerómetro se conecta a las entradas ‘a3’ de los ‘Mod1’, ‘Mod2’, ‘Mod3’. Tal y como se muestra en el diagrama del Anexo 1, Conexionado de los acelerómetros a los cartucho.

A2.3. Sección 3

80	<code>%% Acquire Data</code>	
81	<code>% Start the session in foreground.</code>	
82	<code>[data, timestamps, starttime] = startForeground(s);</code>	
83		
84	<code>%Data contiene los datos en el mismo orden que se han abierto los canales</code>	
85		
86	<code>%% Log Data</code>	
87	<code>% Convert the acquired data and timestamps to a timetable in a workspace variable.</code>	
88	<code>% Primer acelerómetro</code>	
89	<code>% A1X = data(:,1);</code>	
90	<code>% A1Y = data(:,2);</code>	
91	<code>% A1Z = data(:,3);</code>	
92	<code>% DAQ_1 = timetable(seconds(timestamps),A1X,A1Y,A1Z);</code>	
93		
94	<code>% Segundo acelerómetro</code>	
95	<code>% A2X = data(:,4);</code>	
96	<code>% A2Y = data(:,5);</code>	
97	<code>% A2Z = data(:,6);</code>	
98	<code>% DAQ_2 = timetable(seconds(timestamps),A2X,A2Y,A2Z);</code>	
99		
100	<code>% Tercer acelerómetro</code>	
101	<code>% A3X = data(:,7);</code>	
102	<code>% A3Y = data(:,8);</code>	
103	<code>% A3Z = data(:,9);</code>	
104	<code>% DAQ_3 = timetable(seconds(timestamps),A3X,A3Y,A3Z);</code>	
105		
106	<code>% Cuarto acelerómetro</code>	
107	<code>% A4X = data(:,10);</code>	
108	<code>% A4Y = data(:,11);</code>	
109	<code>% A4Z = data(:,12);</code>	
110	<code>% DAQ_4 = timetable(seconds(timestamps),A4X,A4Y,A4Z);</code>	
111		
112	<code>% Quinto acelerómetro</code>	
113	<code>% A5X = data(:,13);</code>	
114	<code>% A5Y = data(:,14);</code>	
115	<code>% A5Z = data(:,15);</code>	
116	<code>% DAQ_5 = timetable(seconds(timestamps),A5X,A5Y,A5Z);</code>	
117		

El comando de la línea 82 se utiliza para crear la tabla en la que se almacenan los datos que estamos recabando. Las columnas de la tabla serán cada uno de los sensores activados. Ya que Matlab ordena las columnas en el orden que se han abierto los canales, toda la parte de 'Log Data' –que sirve para asignar las columnas- lo dejamos como un simple comentario.

A2.4. Sección 4

```

136 %% Plot Data
137 % Plot the acquired data on labeled axes.
138 % plot(DAQ_1.Time, DAQ_1.Variables)
139 % xlabel('Time')
140 % ylabel('Amplitude (V)')
141 % legend(DAQ_1.Properties.VariableNames)
142
143 % Plot A1
144 - figure(1)
145 - plot(seconds(timestamps),data(:,1),seconds(timestamps),data(:,2),seconds(timestamps),data(:,3))
146 % Plot A2
147 - figure(2)
148 - plot(seconds(timestamps),data(:,4),seconds(timestamps),data(:,5),seconds(timestamps),data(:,6))
149 % Plot A3
150 - figure(3)
151 - plot(seconds(timestamps),data(:,7),seconds(timestamps),data(:,8),seconds(timestamps),data(:,9))
152 % Plot A4
153 - figure(4)
154 - plot(seconds(timestamps),data(:,10),seconds(timestamps),data(:,11),seconds(timestamps),data(:,12))
155 % Plot A5
156 - figure(5)
157 - plot(seconds(timestamps),data(:,13),seconds(timestamps),data(:,14),seconds(timestamps),data(:,15))
158 % Plot A6
159 - figure(6)
160 - plot(seconds(timestamps),data(:,16),seconds(timestamps),data(:,17),seconds(timestamps),data(:,18))
161 % Plot A7
162 - figure(7)
163 - plot(seconds(timestamps),data(:,19),seconds(timestamps),data(:,20),seconds(timestamps),data(:,21))
164 % Plot A8
165 - figure(8)
166 - plot(seconds(timestamps),data(:,22),seconds(timestamps),data(:,23),seconds(timestamps),data(:,24))
167
168

```

Desde la línea 144 hasta la 166, hacemos que, una vez terminado el tiempo de toma de datos, salten en pantalla los gráficos que representan el voltaje registrado por las vibraciones respecto al tiempo.

A2.5. Sección 5

```

169 %% Clean Up
170 % Clear the session and channels, if any.
171 - clear s channelA1X channelA1Y channelA1Z channelA2X channelA2Y channelA2Z ...
172         channelA3X channelA3Y channelA3Z channelA4X channelA4Y channelA4Z ...
173         channelA5X channelA5Y channelA5Z channelA6X channelA6Y channelA6Z ...
174         channelA7X channelA7Y channelA7Z channelA8X channelA8Y channelA8Z
175
176
177

```

Por último, en la línea 171 cerramos todos los canales utilizados durante el programa.

ANEXO 3: Gráficos obtenidos

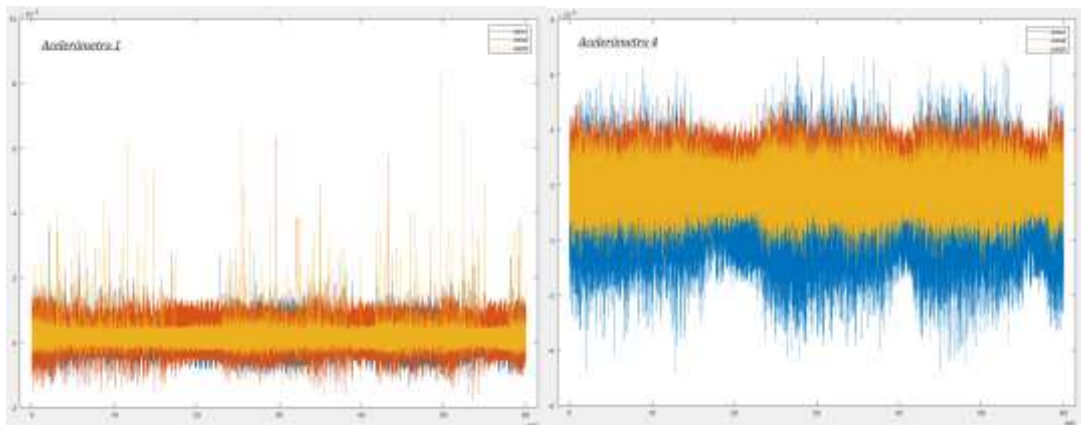
Al final de cada una de las 75 iteraciones, la sección 4 -mostrada en el ANEXO 2- del programa de Matlab nos muestra, en forma de gráficos, las variaciones del voltaje sufridos por los acelerómetros respecto al tiempo. Cada iteración conlleva 8 gráficos diferentes, lo que hace un total de 600 representaciones gráficas.

Por tanto, en vista de dar algunos ejemplos, en este anexo se expondrán únicamente algunos ejemplos, correspondientes a los acelerómetros 1 y 4. El acelerómetro 1 se encuentra justo al lado del excitador, por lo que es un indicador fiable de la vibración entrante. Por otro lado, el acelerómetro 4 está situado en el jacket, concretamente al lado de la barra que se modifica en las diferentes iteraciones.

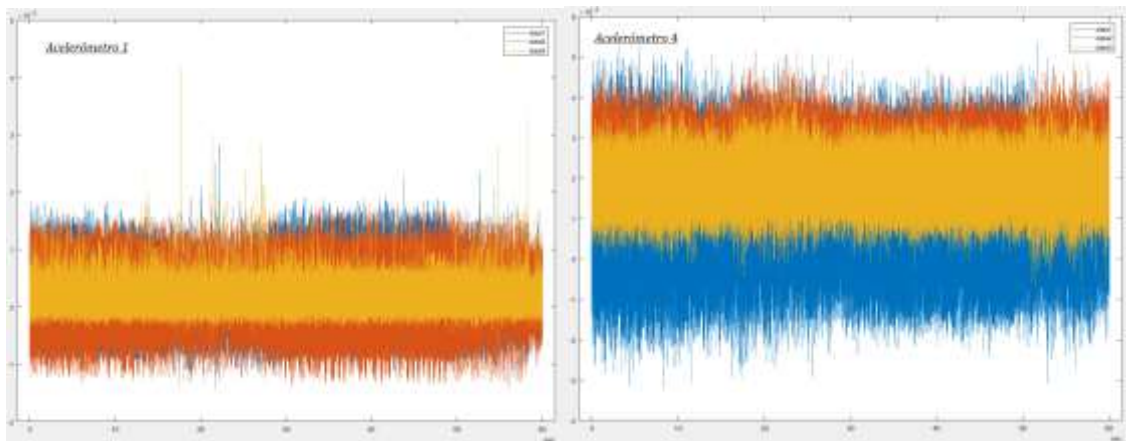
Siguiendo la leyenda que da Matlab a lo hora de exponer el gráfico, la relación es la siguiente: **data1** = componente X; **data2** = componente Y; **data3** = componente Z.

A3.1. Estructura original

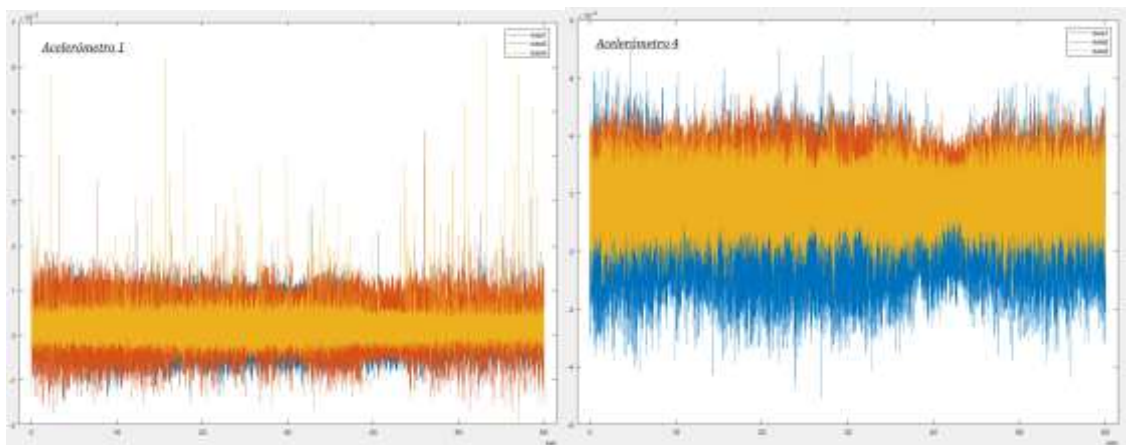
- Amplitud de ruido: **2Vpp**



- Amplitud de ruido: **4Vpp**

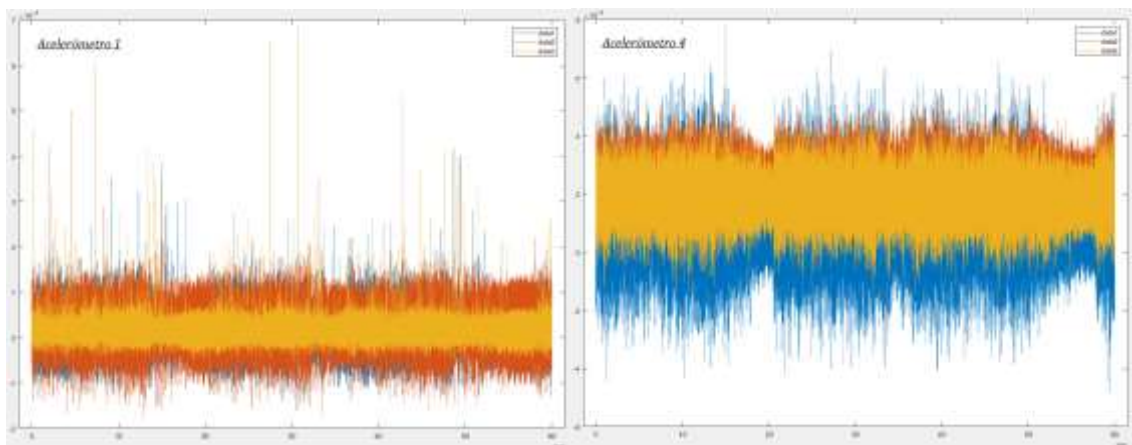


- Amplitud de ruido: **6Vpp**

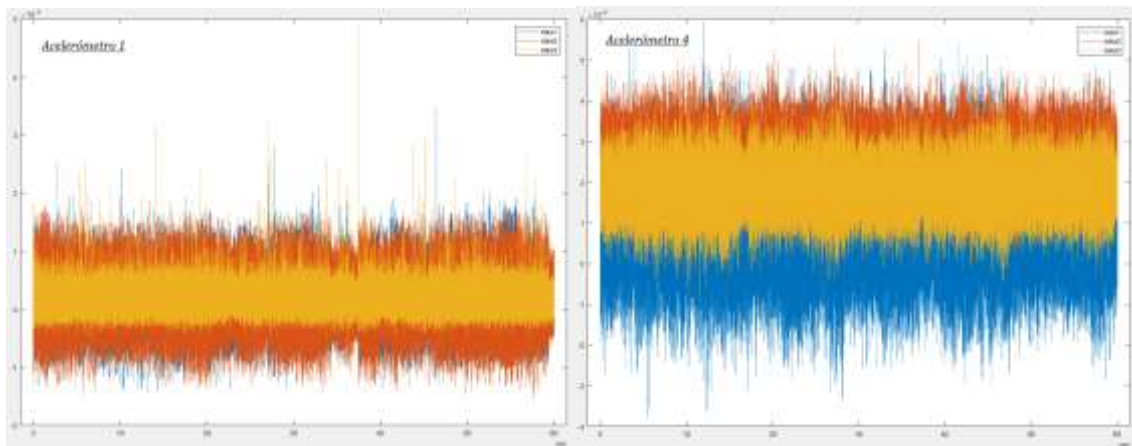


A3.2. Barra seccionada 5mm

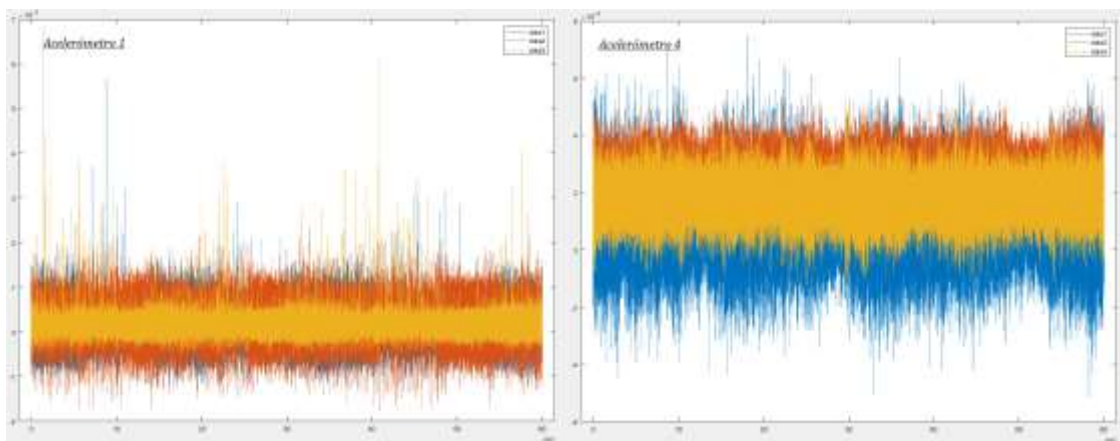
- Amplitud de ruido: **2Vpp**



- Amplitud de ruido: **4Vpp**

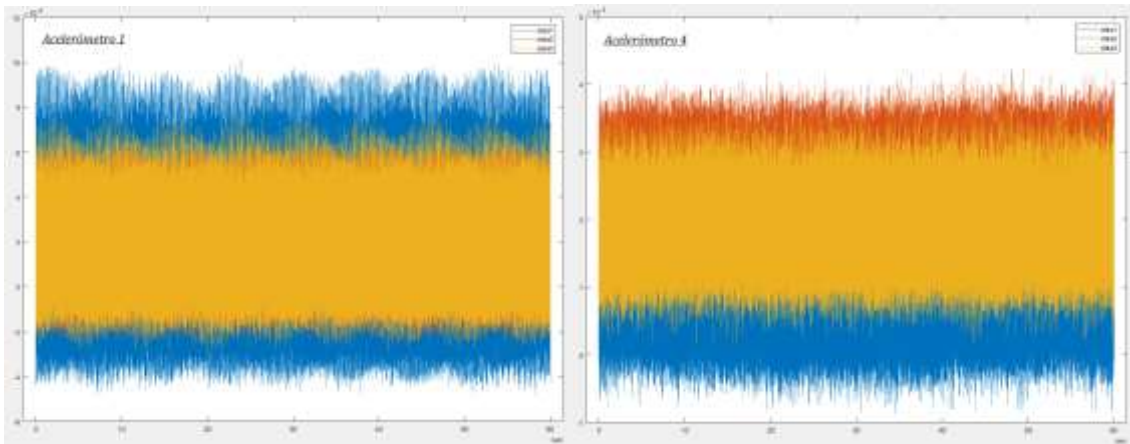


- Amplitud de ruido: **6Vpp**

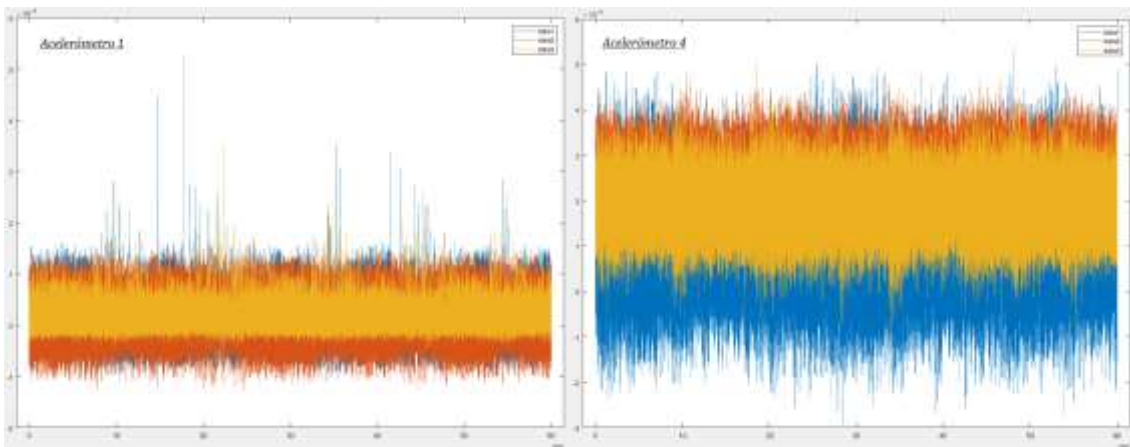


A3.3. Barra sin un tornillo

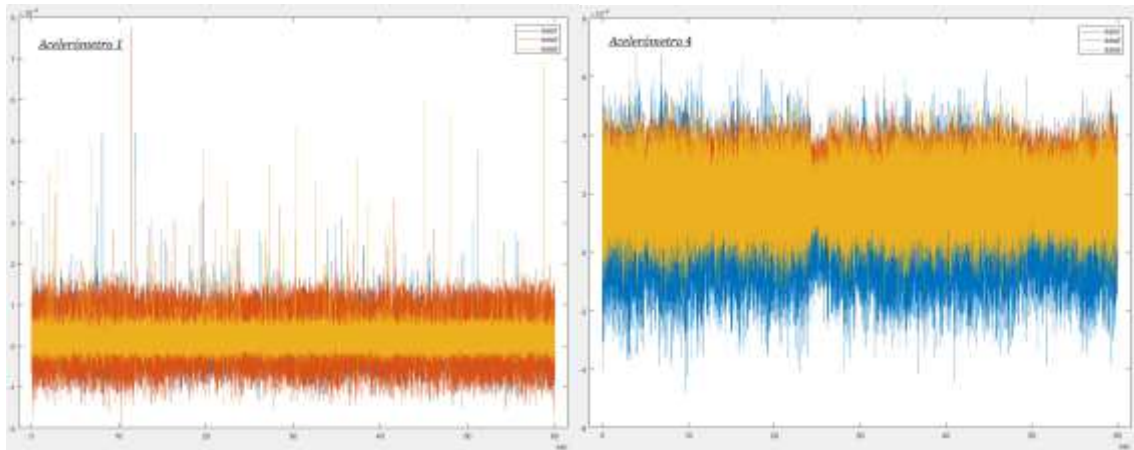
- Amplitud de ruido: **2Vpp**



- Amplitud de ruido: **4Vpp**



- Amplitud de ruido: **6Vpp**



A la hora de interpretar los gráficos es importante tener en cuenta que la escala que se muestra no es fija en todas las imágenes, ya que la señal del excitador no es constante al 100% y ocasionalmente crea picos de voltaje que cambian la escala de las funciones mostradas.



